

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-180127
 (43)Date of publication of application : 26.06.1992

(51)Int.Cl. G06F 9/44
 H04Q 3/545

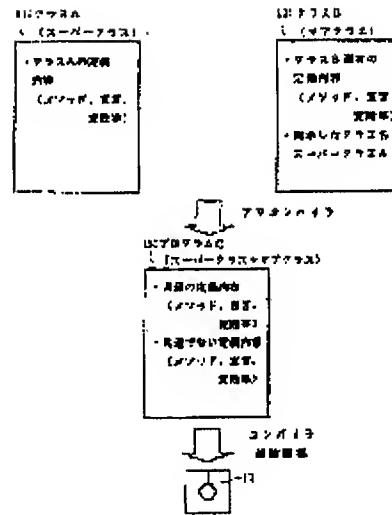
(21)Application number : 02-307212 (71)Applicant : OKI ELECTRIC IND CO LTD
 (22)Date of filing : 15.11.1990 (72)Inventor : ITO NOBUYUKI
 INAFUNE MASAHIRO
 MIZUO GAKUBUN
 ASAI SHIGEYOSHI

(54) METHOD FOR TRANSFORMING CLASS INTO FILE

(57)Abstract:

PURPOSE: To improve the retrieving speed for the definition contents when the object directing software is applied to an exchange system by storing the definition contents of more significant classes that are not duplicated with each other in addition to the definition contents proper to a certain class when this class is transformed into a file.

CONSTITUTION: The classes are transformed into files as follows with the software used in an exchange system which is formed based on the direction of an object. That is, if a class A 11 more significant than a class B 13 exists when the class B 13 is transformed into a file, the definition contents proper to the class B 13 are extracted and then the definition contents of the class A 11 which are not duplicate with each other are taken out. These extracted contents are stored in a program C 15 in the form of a file. Thus the program C 15 storing the definition contents of the class B 13 is retrieved by an access at a high speed even with the definition contents of the class A 11 in a system execution state.



⑯ 公開特許公報 (A) 平4-180127

⑮ Int. Cl. 5

G 06 F 9/44
H 04 Q 3/545

識別記号

330 Z

庁内整理番号

8724-5B
8843-5K

⑯ 公開 平成4年(1992)6月26日

審査請求 未請求 請求項の数 1 (全6頁)

⑯ 発明の名称 クラスのファイル化方法

⑯ 特 願 平2-307212

⑯ 出 願 平2(1990)11月15日

⑯ 発明者	伊藤 信行	東京都港区虎ノ門1丁目7番12号	沖電気工業株式会社内
⑯ 発明者	稻船 正春	東京都港区虎ノ門1丁目7番12号	沖電気工業株式会社内
⑯ 発明者	水尾 学文	東京都港区虎ノ門1丁目7番12号	沖電気工業株式会社内
⑯ 発明者	浅井 重良	東京都港区虎ノ門1丁目7番12号	沖電気工業株式会社内
⑯ 出願人	沖電気工業株式会社	東京都港区虎ノ門1丁目7番12号	
⑯ 代理人	弁理士 鈴木 敏明		

明細書

1. 発明の名称

クラスのファイル化方法

2. 特許請求の範囲

情報とその情報を取扱う手続きを一体化したオブジェクト指向と呼ばれる概念に基づいて作成される交換システム用ソフトウェアにおいて、

一のクラスをファイル化する際、その一のクラスに継承した上位クラス名が存在する場合、

前記一のクラスの固有の定義内容を取り出し、次に継承元の上位クラスの定義内容のうち、既に取出した定義内容と重複しないものを取り出し、重複した定義内容についてはその整合性を確認し、これらの取出した定義内容を前記一のクラスの定義内容としてメモリにファイル化して格納するようにしたことの特徴とするクラスのファイル化方法。

3. 発明の詳細な説明

(産業上の利用分野)

本発明は、P BXなどの交換システム用オブ

ジェクト指向ソフトウェアにおけるクラスのファイル化方法に関する。

(従来の技術)

現在、交換システム用ソフトウェアの開発には膨大なコストと期間を必要としている。また、既に開発された交換システム用ソフトウェアに新しい機能やサービスを追加するときも同様である。

そこで、交換システム用ソフトウェアの開発力を向上させるためには、ソフトウェアを再利用可能なソフトウェア部品として蓄積することが必要である。そして、部品化したソフトウェアを再利用したり、新たなソフトウェア部品を追加したりすることで、新しいソフトウェア部品を少ない設計量で生産することができ、ソフトウェアの生産性の向上を図ることができる。

一方ソフトウェア部品を組合せることで、種々のソフトウェア交換システムが構築できなければならない。交換システムを構築するためには、交換システムを、人がより理解し易い「要素」相互の関係としてモデル化しなければならない。従つ

て、ソフトウェア部品は、交換システムモデルの構成「要素」と対応していることが重要である。

交換システムモデルの要素を定義する方法として、ファンクション指向方法 (Function Oriented Method) とオブジェクト指向方法 (Object Oriented Method) がある。前者の方法は、システムが持つべき機能を F E (Functional Element) として、定義する作業と、外部のシステムからのアクセスに応じて実行されて一連の F E 群を F C (Functional Component) として定義する作業から成り立っている。後者の方法は、前記 F E や F C の対象となるユーザー、端末、トランク、呼び、ルートなどを、オブジェクトの種類 (クラス) として定義し、F E や F C は、各クラスの実体 (インスタンス) となるオブジェクトにカプセル化されていると考える。

しかし、ソフトウェア部品を F E や F C に対応して定義すると、ソフトウェア部品のサイズが小さすぎて再利用価値に乏しくなり、またソフトウェア部品によるシステムの組立作業が大幅に複

雑化するなど種々の問題を有する。このためソフトウェア部品をオブジェクト指向方法で定義される“クラス”及び“オブジェクト”に対応させて定義することが必要である。ここに、オブジェクトとは、情報とその情報を取扱うための手続き (メソッド) を一体化 (カプセル化) したものであり、オブジェクトにアクセスするには、そのオブジェクトにメッセージを送る以外に方法がなく、オブジェクト自身がメッセージを解釈し対応する機能を実行する。又、クラスとは、オブジェクトが管理する情報と個々のメッセージに対する手続きを定義したものである。

交換システム用オブジェクト指向ソフトウェアの生産性の向上を図るために、クラスの継承が行なわれている。インヘリタンスは、クラス定義の相互の間での定義の継承ができる機構であり、この機構により他のクラスの定義をベースに定義の差分 (追加、差し替え) を与えることで、新しいクラスを作ることができる。従って、インヘリタンスにクラス A (スーパークラス A) とクラス B

(サブクラス B) を入力してやると、出力としてスーパークラス A を継承したクラス B を得ることができる。この得られたクラス B には、継承元であるスーパークラス (上位クラス) A のすべての宣言とクラス定義の操作 (メソッド) が引き継がれる。しかし、下位クラスであるクラス B (サブクラス B) には、第 2 図に示すようにクラス固有の定義内容 (宣言、メソッド等) が書かれている他、継承されたスーパークラス A の名称が書かれているだけで、スーパークラス A の具体的な定義内容は書かれていらない。尚、第 2 図はスーパークラスとそのスーパークラスを継承したサブクラスの関係を示す説明図である。

そして、従来、クラスをファイル化する場合、下位クラス (サブクラス) のクラス B とその継承元の上位クラス (スーパークラス) であるクラス A とを、それぞれ別々にファイル化していた。

従って、交換システム用オブジェクト指向のソフトウェアを交換システムに適用した場合、システムのプログラムの実行時に、下位クラスである

クラス B のファイルをアクセスし、そのファイルのクラス B の中で例えば該当するメソッドを検索し、該当するメソッドがなかったら、継承している上位クラスであるクラス A のファイルをアクセスし、そのファイルのクラス A の中より該当するメソッドを検索し取出すことになる。この時、始めて上位クラスであるクラス A の定義内容中の該当するメソッドが、下位クラスであるクラス B に実質的に継承されたことになる。即ち、プログラムの実行時に継承する動的継承である。

尚、クラスの定義は、クラス名の宣言、インスタンスの宣言、変数の宣言、メソッド.i. (i = 1 ~ n) の定義から構成されている。また、クラスの継承についての関係文献として、「オブジェクト指向プログラミングソフトウェア開発技術の進化」(特に第 5 節、継承: その実現方法 (P. 93 ~ P. 99))、著者: B. J コックス、監訳者: 前川守、発行所: トッパンがある。

(発明が解決しようとする課題)

上述した従来のクラスのファイル化方法では、

下位クラスが上位クラスを継承していても、下位クラスにはその継承した上位クラス名しか書かれていないので、システム運転時に、下位クラスのファイルをアクセスし、例えば該当するメソッドがなかったら継承元である上位クラスのファイルをアクセスして、その上位クラスの定義内容を検索していた。

従って、従来のオブジェクト指向ソフトウェアを大規模な交換システムに適用する場合、従来のクラスのファイル化方法では、次のような問題点を有する。

(1) 下位クラスのファイルをアクセスし、その下位クラスの中で、例えば該当するメソッドを検索し、該当するメソッドがなかったら、その後、継承元の上位クラスのファイルをアクセスし、その上位クラスのメソッドを検索することになり、2つのファイルをアクセスすることになり、余計な時間がかかる。従って、メソッド等のクラスの定義内容を検索する時のスピードが遅くなる。

(2) 型の異なる同一名称の変数（例えば整数型

の変数 a とキャラクタ型の変数 a ）や、同一タグ名（構造体の名前）でありながらメモリのメンバー構成（構造体の中の変数名の構成）が異なる構造体等があると、下位クラスとその上位クラスとをそれぞれ別々にメモリファイル化しているため下位クラス（サブクラス）と上位クラス（スーパークラス）での解釈が異なり、下位クラスのプログラムと、上位クラスのプログラムの整合性がうまくいかず、プログラム実行時にプログラムが異常動作をしてシステムが動かないということが起こる可能性がある。つまり、下位クラスとその継承元の上位クラスとを別々にファイル化したのでは、プログラムの整合性がとれず、信頼性が低くなる。

そこで、本発明の目的は、交換システム用オブジェクト指向ソフトウェアを交換システムに適用し、システム実行時に継承元の上位クラスの定義内容でも高速に検索することができ、高信頼性が維持できるクラスのファイル化方法を提供することにある。

（課題を解決するための手段）

本発明のクラスのファイル化方法は、情報とその情報を取扱う手続きを一体化したオブジェクト指向と呼ばれる概念に基づいて作成される交換システム用ソフトウェアにおいて、一のクラスをファイル化する際、その一のクラスに継承した上位クラス名が存在する場合、前記一のクラスの固有の定義内容を取出し、次に継承元の上位クラスの定義内容のうち、既に取出した定義内容と重複しないものを取出し、これらの取出した定義内容を前記一のクラスの定義内容としてメモリにファイル化して格納するようにしたものである。

（作用）

本発明のクラスのファイル化方法では、一のクラスをファイル化する際にその一のクラスに継承した上位クラスの名が存在する場合、前記一のクラスの固有の定義内容をまず取出し、次に継承元の上位クラスの定義内容のうち重複しないものを取出し、これらの取出した定義内容を前記一のクラスの定義内容としてメモリにファイル化して格

納するようにしたので、交換システム用オブジェクト指向ソフトウェアを交換システムに適用し、システム実行時に、継承元の上位クラスの定義内容でも、一のクラスの定義内容を格納したファイルと同じファイルをアクセスすることにより高速に検索することができ、高信頼性を維持できる。

（実施例）

本発明は、交換システム用オブジェクト指向ソフトウェアにおけるクラス（スーパークラスを継承したクラス）のファイル化方法に関するものである。

本発明の実施例について第1図を用いて説明する。

第1図は本発明によるクラスのファイル化方法の一実施例を示す説明図である。

同図において、11はフロッピーディスク等のメモリに格納されたスーパークラス（上位クラス）としてのクラスAであって、このクラスA11には、クラスAの定義内容（宣言、変数、メソッド等）が書かれている。また、13は、フ

ロッピーディスク等のメモリに格納されたサブクラス（下位クラス）としてのクラスBであって、このクラスB 1 3には、クラスB固有の定義内容（宣言、変数、メソッド等）と継承したスーパークラス（クラスA）の名が書かれている。尚、クラスAとクラスBは、オブジェクト指向プログラミング言語で書かれたプログラムである。また、1 5はクラスBとその継承元であるスーパークラスA（クラスA）より重複しないように取出された定義内容が、クラスAとクラスBに共通の定義内容と共通でない定義内容とに分けられ、汎用プログラミング言語（C言語）で書かれたプログラム（区別する便宜上、プログラムCという。）であって、このプログラムCはメモリに格納される。このプログラムCをコンバイラに入力することにより、機械語プログラムに展開し、プログラムA、Bで参照する他のプログラム（ライブラリ等）を連結編集することにより、コンピュータで実行可能なプログラム（ロードモジュール）を作成し、フロッピーディスク1 7に格納する。尚、

をリードし）、既にリードされて内部メモリに蓄積された情報と比較して、そのリードした変数やメソッドや構造体宣言などの定義内容に矛盾が生じないか否か（重複がないか否か）をチェックしながら重複しないものを内部メモリに記憶する。この場合、プリコンバイラはスーパークラスAの定義内容にクラスBの定義内容と同一名のメソッド、変数、宣言等が現われたときは、下位クラスであるクラスBのものを優先して内部メモリに記憶する。そして、継承元のスーパークラスの全て（ここでは、継承元のスーパークラスはクラスAのみであるので、クラスAのみをいう。）について上記処理を終えたら、プリコンバイラは、内部メモリに記憶しておいた定義内容をクラスBとスーパークラスAとに共通な定義内容と共通でない定義内容とに分けてC言語のプログラムC 1 5を作成し、これを外部メモリに格納する。

次にCPUは、外部メモリに格納された定義内容（重複せずに取出したクラスBとその継承元のスーパークラスAの定義内容）を取出して、メモ

1 7はメモリとしてのフロッピーディスクであって、このフロッピーディスク1 7にクラスBのファイル化が成される。

次にスーパークラスAを継承したクラスBのファイル化について説明する。まず、クラスBとその継承元であるスーパークラスAをプリコンバイラに入力して、プログラムCを作成する。ここで、プリコンバイラとは、オブジェクト指向言語で記述されたソースプログラムを汎用プログラミング言語（C言語）のソースプログラムに変換する装置である。

そこで、クラスAを格納したメモリとクラスBを格納したメモリをプリコンバイラの入力側に接続してやると、プリコンバイラは、まずメモリに格納されているファイル化すべきクラスBをリードし、そのクラスB固有の定義内容である宣言、変数、メソッド等を内部メモリに記憶する、次にプリコンバイラは、クラスBの継承元のスーパークラスを継承順にリードし（ここでは、継承元はスーパークラスAだけであるので、クラスAのみ

リとしての、例えばフロッピーディスク1 7に、クラスBの定義内容としてファイル化を行なう。これによりフロッピーディスク1 7に、ファイル化したクラスBの定義内容には、スーパークラスAの名称でなくそのスーパークラスAの定義内容も含まれることになる。

以上のようにして、交換システム用オブジェクト指向ソフトウェアにおいて、スーパークラスを継承したクラスをファイル化すれば、ファイル化時に、継承元のスーパークラスの定義内容（メソッド、宣言、変数など）の継承も行なっている。従って、交換システム用オブジェクト指向ソフトウェアを交換システムに適用した場合、システム実行時に、クラスBの定義内容を格納したフロッピーディスク1 7のファイルをアクセスすれば、クラスBの継承元のスーパークラスAの定義内容も同一ファイルに格納されているので、同一ファイルの中を検索するだけで、スーパークラスAの該当する定義内容も取出すことができる。よって、システム実行時に、従来の如き、継承元

であるスーパークラス A の継承処理を行なう必要がなくなり、その分該当する定義内容の検索スピードを従来に比べ上げることができる。また、クラス B のファイル化時に、予め継承元のスーパークラス A の定義内容がクラス B 固有の定義内容と重複しないように継承の整合性のチェックを行なっているので、オブジェクト指向ソフトウェアで構築した交換システムの実行時に、プログラムが従来の如き異常動作をしたりせず、信頼性の向上が期待できる。

本実施例においては、クラス B がスーパークラス A のみを継承した場合であるが、クラス B が複数スーパークラスを継承した場合でも、同様に本発明を適用できる。この場合、メモリにファイル化したクラス B の定義内容には、クラス B 固有の定義内容の他、その継承元の複数のスーパークラスの定義内容も含まれることになる。但し、ファイル化された複数のスーパークラスの定義内容は、継承順に重複しないように本実施例で説明した如く取出された定義内容である。

と重複しないように継承の整合性のチェックを行なっているので、オブジェクト指向ソフトウェアで構築した交換システム（大規模な交換システムも含む）の実行時に、プログラムが従来の如き異常動作をしたりせず、信頼性の向上を図ることができる。

4. 図面の簡単な説明

第 1 図は本発明によるクラスのファイル化方法の一実施例を示す説明図、第 2 図はスーパークラスとサブクラスの関係を示す説明図である。

1 1 … クラス A、1 3 … クラス B、

1 5 … プログラム C、

1 7 … フロッピーディスク。

特許出願人 沖電気工業株式会社

代理人 鈴木 敏明

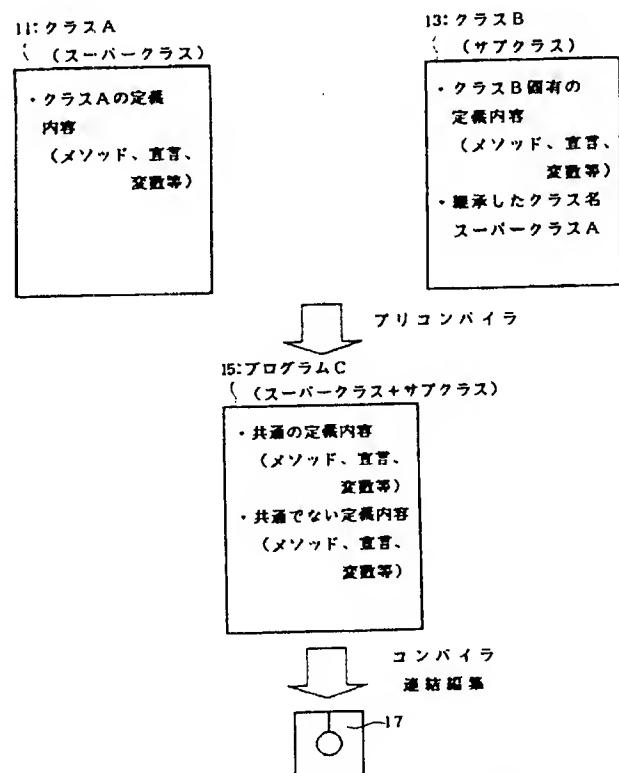
本発明は、本実施例に限定されることなく、本発明の要旨を逸脱しない範囲で、種々の応用及び変形が考えられる。

(発明の効果)

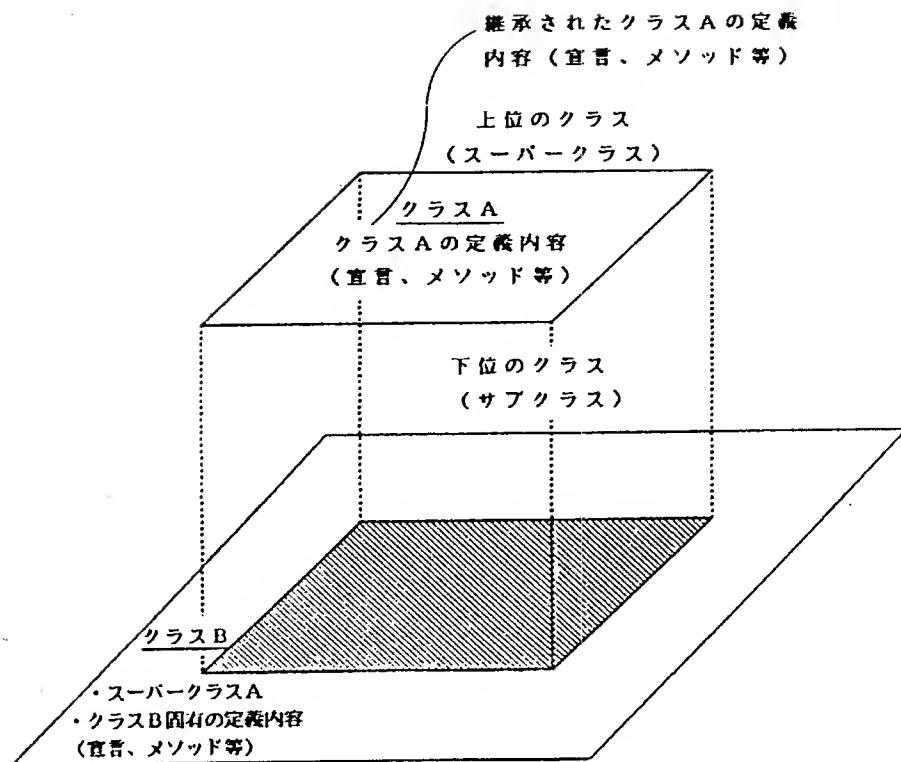
上述したように本発明によれば次のような効果が得られる。

(1) 同一ファイルに、一のクラスの定義内容として、継承元の上位クラスの定義内容も含まれているので、交換システム用オブジェクト指向ソフトウェアを交換システム（大規模な交換システムも含む）に適用した場合、システム実行時に、一のクラスの定義内容を格納した同一ファイルの中を検索するだけで、継承元の上位クラスの該当する定義内容も取出すことができる、従って、システム実行時に、従来の如き継承元である上位クラスの継承処理を行なう必要がなくなり、その分該当する定義内容の検索スピードを従来に比べ上げることができる。

(2) 一のクラスのファイル化時に、予め継承元の上位クラスの定義内容が既に取出した定義内容



本発明のクラスのファイル化方法の一実施例
第 1 図



スーパークラスとサブクラスの関係

第二回